WHAT IS CLAIMED IS:

1. A processing core that executes a compare instruction, the processing core comprising:

a plurality of general-purpose registers comprising a first input operand register, a second input operand register and an output operand register;

a register file comprising the plurality of general-purpose registers;

comparison logic coupled to the register file, wherein the comparison logic tests for at least two of the following relationships: less than, equal to, greater than and no valid relationship;

decode logic which selects the output operand register from the plurality of general-purpose registers; and

a store path between the comparison logic and the selected output operand register.

2. The processing core that executes the compare instruction as set forth in claim 1, wherein a very long instruction word includes a plurality of compare instructions.

3. The processing core that executes the compare instruction as set forth in claim 1, wherein decode logic selects the first and second input operand registers from the plurality of general-purpose registers.

4. The processing core that executes the compare instruction as set forth in claim 1, wherein the processing core issues a plurality of compare instructions at one time.

5. The processing core that executes the compare instruction as set forth in claim 1, further comprising:

a first load path between the first input operand register and comparison logic; and

a second load path between the second input operand register and comparison logic.

6. The processing core that executes the compare instruction as set forth in claim 1, wherein the output operator register stores a value indicating a

relationship between the first and second input operator registers which is at least one of greater than, less than, equal to and not a number.

7. The processing core that executes the compare instruction as set forth in claim 6, wherein the not a number value indicates a comparison between the first and second input operands that cannot be made.

8. The processing core that executes the compare instruction as set forth in claim 6, wherein the value is an integer.

9. The processing core that executes the compare instruction as set forth in claim 1, wherein:

the first input operand register is a double precision floating point data type;

the second input operand register is a single precision floating point data type; and

the output operand register is a double precision floating point data type.

10. The processing core that executes the compare instruction as set forth in claim 1, further comprising a plurality of processing paths that are coupled to the register file.

11. The processing core that executes the compare instruction as set forth in claim 1, wherein the register file comprises special purpose registers which cannot store an output operand.

12. A method for performing a compare operation, the method comprising steps of:

decoding a compare instruction;

configuring first and second paths between a register file and comparison logic;

configuring a third path between the comparison logic and the register file;

comparing a first input operand and a second input operand to produce a result which indicates an absence of at least three mathematical relationships between the first input operand and the second input operand; and

coupling an output operand to a general-purpose register in the register file.

13. The method for performing the compare operation as set forth in claim 12, the method further comprising a step of enabling the comparison logic in an arithmetic logic unit.

14. The method for performing the compare operation as set forth in claim 12, wherein the configuring steps each comprise a step of addressing a general-purpose register in the register file.

15. The method for performing the compare operation as set forth in claim 12, wherein a very long instruction word comprises the compare operation.

16. The method for performing the compare operation as set forth in claim 12, wherein the comparing step comprises a step of converting a data type of at least one of the first and second input operands.

17. A method for executing a compare instruction in a processor, the method comprising steps of:

issuing the compare instruction;

comparing a first input operand and a second input operand to determine at least two mathematical relationships between the first and second input operands;

determining an output operand indicative of the mathematical relationships; and

storing the output operand in a general-purpose register of a register file.

18. The method for executing the compare instruction in the processor as set forth in claim 17, wherein the comparing step comprises:

determining if the first input operand is less than the second input operand;

determining if the first input operand is greater than the second input operand;

determining if the first input operand is equal to the second input operand; and

determining if there is no valid relationship between the first input operand and the second input operand.

19. The method for executing the compare instruction in the processor as set forth in claim 17, wherein the compare instruction is a very long instruction word which comprises a plurality of compare instructions which are processed in parallel down separate processing paths.

20. The method for executing the compare instruction in the processor as set forth in claim 17, wherein the general-purpose register is used to store operators from other types of instructions.